UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/598,436	06/22/2000	Charles Robert Moore	AT9-99-453	5581

7590 07/15/2003

BRACEWELL & PATTERSON LLP
Intellectual Property Law
P O Box 969
Austin, TX 78767-0969

EXAMINER
HUISMAN, DAVID J

ART UNIT	PAPER NUMBER
2183	4

DATE MAILED: 07/15/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Offic Action Summary	Applicati n No. 09/598,436	Applicant(s) MOORE, CHARLES ROBERT
	Examiner David J. Huisman	Art Unit 2183

-- Th MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 June 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☒ Claim(s) *1,2 and 13* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 June 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *2* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-16 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: #2. IDS as received on 9/15/2000 and #3. Change of Address as received on 7/22/2002.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. The examiner recommends specifying the types of operations that result from the load split (prefetch and register) within the title.

4. The disclosure is objected to because of the following informalities: In TABLE I on page 11, In the boxes listed "Cycle 3" and "Cycle 5," the word "Cycle" should appear as one word instead of being separated and continued on a second line (note the last letter of "Cycle" is on a separate line)

Appropriate correction is required.

Claim Objections

5. Claim 1 is objected to because of the following informalities: In line 8, change "need" to --needed--. Appropriate correction is required.

6. Claim 2 is objected to because of the following informalities: In line 7, change "need" to --needed--. Appropriate correction is required.

7. Claim 13 is objected to because of the following informalities: In line 2, should "load operation" be changed to --said load instruction--. Also, line 3 should be modified such that it sounds more grammatically correct. For instance, "operation have" should be changed to --operations having--. Or, line 3 could be replaced with --into a prefetch operation and a register operation that have a same--. Appropriate correction is required.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-2, 10-11, and 13-15 are rejected under 35 U.S.C. 102(b) as being anticipated by DeLano et al., U.S. Patent No. 5,396,604 (herein referred to as DeLano).

10. Referring to claim 1, DeLano has taught a method of processing an instruction in a processor, said method comprising:

a) fetching a sequence of instructions including an instruction. It is inherent that a processor fetches a sequence of instructions so that they can be executed. Fig.2 shows a sequence of instructions which would be fetched by the processor.

b) translating the instruction into separately executable prefetch operation and register operations, wherein said prefetch operation obtains, in an out-of-order fashion, data needed to

execute said register operation and said register operation performs an operation in order. See column 3, lines 1-6, and column 5, lines 20-24, and note that the compiler will insert a prefetch instruction (LOAD_to_GR0) prior to the load instruction that accesses the same memory location. This increases the likelihood of the actual load instruction hitting the cache, thereby masking memory latency. From column 5, lines 20-24, it can be seen that the prefetch instruction is inserted out-of-order (with respect to the actual load) some X instructions before the actual in-order load instruction.

11. Referring to claim 2, DeLano has taught a processor comprising:

a) means for fetching a sequence of instructions including an instruction. It is inherent that a processor fetches a sequence of instructions so that they can be executed. Fig.2 shows a sequence of instructions which would be fetched by the processor.

b) means for translating the instruction into separately executable prefetch operation and register operations, wherein said prefetch operation obtains, in an out-of-order fashion, data needed to execute said register operation and said register operation performs an operation in order. See column 3, lines 1-6, and column 5, lines 20-24, and note that the compiler will insert a prefetch instruction (LOAD_to_GR0) prior to the load instruction that accesses the same memory location. This increases the likelihood of the actual load instruction hitting the cache, thereby masking memory latency. From column 5, lines 20-24, it can be seen that the prefetch instruction is inserted out-of-order (with respect to the actual load) some X instructions before the actual in-order load instruction.

12. Referring to claim 10, DeLano has taught a method of performing a load operation in a processor having a plurality of registers, said method comprising:

a) fetching a load instruction and a preceding instruction that precedes said load instruction in program order. See column 3, lines 1-6, and column 5, lines 20-24, and note that the processor will fetch a load instruction and a preceding instruction, where the preceding instruction would be considered one of the "Other instructions" shown in column 5. This fetching must inherently occur if the instructions are to be executed.

b) detecting said load instruction and translating said load instruction into separately executable prefetch and register operations. See column 3, lines 1-6, and column 5, lines 20-24, and note that a prefetch instruction (LOAD_to_GR0) is inserted prior to the load instruction that accesses the same memory location. This increases the likelihood of the actual load instruction hitting the cache, thereby masking memory latency.

c) performing at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data. From column 5, lines 20-24, it can be seen that the prefetch instruction is inserted out-of-order (with respect to the actual load) some X instructions before the actual in-order load instruction. Consequently, it can be seen that the prefetch instruction would be executed before one or more preceding instructions with respect to the load instruction (as denoted by "Other instructions" in the aforementioned passage). On another note, this instruction can be executed anywhere out-of-order prior to the load instruction, but to be most effective, it should be executed early enough such that the prefetch will finish before the register operation occurs and late enough so that the prefetched data isn't replaced in the cache before the register operation is performed.

d) thereafter, separately executing said register operation to place said data into a register among said plurality of registers specified by said load instruction. See column 5, lines 20-24, and note

that first a prefetch is executed, then some additional instructions are executed, and finally the register load operation is performed.

13. Referring to claim 11, DeLano has taught a method as described in claim 10. DeLano has further taught executing said register operation in-order with respect to said preceding instruction. See column 5, lines 20-24, and note that the register operation (LOAD from address A to GRX) is executed after the preceding "Other instructions" (in-order) so that the prefetch operation has time to complete.

14. Referring to claim 13, DeLano has taught a processor as described in claim 10. DeLano has further taught that translating said load instruction comprises translating load operation into prefetch and register operations having a same operation code. See column 5, lines 20-24, and column 2, lines 48-50. It should be realized that a prefetch is in fact a load operation (just like the actual load-register operation). The load is viewed by the system as a prefetch instruction only when register 0 is specified within the instruction.

15. Referring to claim 14, DeLano has taught a processor as described in claim 13. DeLano has further taught that said prefetch operation and said register operation differ only in a value of a register operation field. Recall from the rejection of claim 13, that the prefetch and register operations differ only in the register operation field since only the prefetch operation will specify register 0.

16. Referring to claim 15, DeLano has taught a processor as described in claim 10. DeLano has further taught that performing said prefetch operation comprises storing said data in a temporary register. See column 5, lines 39-49. DeLano has disclosed that prefetched data can be stored in some general register GRX, where X is not equal to 0.

Claim Rejections - 35 USC § 103

17. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

18. Claims 3-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over DeLano, as applied above, in view of Tanenbaum, "Structured Computer Organization, 2nd Edition", 1984 (herein referred to as Tanenbaum).

19. Referring to claim 3, DeLano has taught a processor comprising:

a) a plurality of registers. See column 3, lines 9-12, and TABLE 1.

b) fetching a load instruction and a preceding instruction that precedes said load instruction in program order. See column 3, lines 1-6, and column 5, lines 20-24, and note that the processor will fetch a load instruction and a preceding instruction. This fetching must inherently occur if the instructions are to be executed.

c) DeLano has not taught that instruction processing circuitry (hardware), responsive to detecting said load instruction, translates said load instruction into separately executable prefetch and register operations. Instead, DeLano has taught that the compiler (software) is used to translate a load instruction such that a prefetch instruction (LOAD_to_GR0) is inserted prior to the load instruction that accesses the same memory location. See column 3, lines 1-6, and column 5, lines 20-24, and note that this increases the likelihood of the actual load instruction hitting the cache, thereby masking memory latency. Tanenbaum has taught that software and hardware are logically equivalent and that any operation performed by software can also be

performed by hardware, and that the choice between the two may be based on the designer's goals. See page 11. Furthermore, a person of ordinary skill in the art at the time of the invention would have recognized that one reason to have actual hardware performing the translation, as opposed to the compiler, is that it would allow old programs, compiled without benefit of the prefetch operation insertion to still benefit from the hardware's ability to perform prefetching. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to use instruction processing circuitry (hardware) to translate a load instruction in a manner described above.

c) DeLano has further taught execution circuitry that performs at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data and subsequently separately executes said register operation to place said data into a register among said plurality of registers specified by said load instruction. From column 5, lines 20-24, it can be seen that the prefetch instruction is inserted out-of-order (with respect to the actual load) some X instructions before the actual in-order load instruction. Consequently, it can be seen that the prefetch instruction would be executed before one or more preceding instructions with respect to the load instruction (as denoted by "Other instructions" in the aforementioned passage).

20. Referring to claim 4, DeLano in view of Tanenbaum has taught a processor as described in claim 3. DeLano has further taught that said execution circuitry executes said register operation in-order with respect to said preceding instruction. See column 5, lines 20-24, and note that the register operation (LOAD from address A to GRX) is executed after the preceding "Other instructions" (in-order) so that the prefetch operation has time to complete.

21. Referring to claim 5, DeLano in view of Tanenbaum has taught a processor as described in claim 3. DeLano has not explicitly taught that said execution circuitry executes said register operation out-of-order with respect to said preceding instruction. However, Official Notice is taken that out-of-order execution and its advantages are well known and expected in the art. With out-of-order execution, instructions can be executed as soon as their data operands are available, as opposed to in-order execution, where if a preceding instruction is stalled, then all subsequent instructions are stalled as well. Therefore, it can be seen that if the preceding instruction is stalled, out-of-order execution would allow the register operation to continue without stalling, thereby maintaining throughput. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to execute the register operation out-of-order with respect to the preceding instruction.

22. Referring to claim 6, DeLano in view of Tanenbaum has taught a processor as described in claim 3. DeLano has further taught that said prefetch operation and said register operation have a same operation code. See column 5, lines 20-24, and column 2, lines 48-50. It should be realized that a prefetch is in fact a load operation (just like the actual load-register operation). The load is viewed by the system as a prefetch instruction only when register 0 is specified within the instruction.

23. Referring to claim 7, DeLano in view of Tanenbaum has taught a processor as described in claim 6. DeLano has further taught that said prefetch operation and said register operation differ only in a value of a register operation field. Recall from the rejection of claim 6, that the prefetch and register operations differ only in the register operation field since only the prefetch operation will specify register 0.

24. Referring to claim 8, DeLano in view of Tanenbaum has taught a processor as described in claim 3. DeLano has further taught that said execution circuitry stores said data prefetched in response to said prefetch operation in a temporary register. See column 5, lines 39-49. DeLano has disclosed that prefetched data can be stored in some general register GRX, where X is not equal to 0.

25. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over DeLano in view of Tanenbaum, as applied above, in view of Konigsburg et al., U. S. Patent No. 5,931,957 (herein referred to as Konigsburg).

26. Referring to claim 9, DeLano in view of Tanenbaum has taught a processor as described in claim 3. DeLano in view of Tanenbaum has not explicitly taught a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation. However, Konigsburg has taught that speculative instructions along a mispredicted branch path must be flushed and their associated results discarded since any changes made to data by these instructions would cause a data hazard in that the data modified by these instructions should not be accessed by subsequent instructions. See column 6, line 66, to column 7, line 7. This type of flushing and discarding is well known and expected in the art and a person of ordinary skill in the art would have recognized that if a prefetch instruction and the associated load instruction are along a mispredicted branch path, then the data loaded by the instructions is undesired, and therefore, should be discarded. Also, the instructions should be cancelled so that they cannot complete and make any undesired changes to the system. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the

invention to implement, within DeLano's system, a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation.

27. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over DeLano, as applied above.

28. Referring to claim 12, DeLano has taught a method as described in claim 10. DeLano has not explicitly taught that executing said register operation out-of-order with respect to said preceding instruction. However, Official Notice is taken that out-of-order execution and its advantages are well known and expected in the art. With out-of-order execution, instructions can be executed as soon as their data operands are available, as opposed to in-order execution, where if a preceding instruction is stalled, then all subsequent instructions are stalled as well. Therefore, it can be seen that if the preceding instruction is stalled, out-of-order execution would allow the register operation to continue without stalling, thereby maintaining throughput. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to execute the register operation out-of-order with respect to the preceding instruction.

29. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over DeLano, as applied above, in view of Konigsburg, as applied above.

30. Referring to claim 16, DeLano has taught a method as described in claim 10. DeLano has not explicitly taught detecting a data hazard for said data, and in response to detection of said hazard for said data, discarding said data and said register operation. However, Konigsburg has

taught that speculative instructions along a mispredicted branch path must be flushed and their associated results discarded since any changes made to data by these instructions would cause a data hazard in that the data modified by these instructions should not be accessed by subsequent instructions. See column 6, line 66, to column 7, line 7. This type of flushing and discarding is well known and expected in the art and a person of ordinary skill in the art would have recognized that if a prefetch instruction and the associated load instruction are along a mispredicted branch path, then the data loaded by the instructions is undesired, and therefore, should be discarded. Also, the instructions should be cancelled so that they cannot complete and make any undesired changes to the system. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement, within DeLano's system, a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation.

Conclusion

31. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Babaian et al., U.S. Patent No. 5,889,985, has taught an array prefetch apparatus and method. It is disclosed that the processor interprets a load instruction as a prefetch and a register operation.

Arora et al, U.S. Patent No. 5,948,095, has taught a method and apparatus for prefetching data in a computer system. The prefetch instruction coming before the corresponding load instruction includes an indication of whether or not it is to ignore exceptions, if one occurs.

Eickemeyer et al., U.S. Patent No. 5,377,336, has taught an improved method to prefetch load instruction data.

Afsar et al., U.S. Patent No. 6,401,193, has taught dynamic data prefetching based on a program counter and addressing mode. More specifically, prefetching is done based on the value in the program counter which would locate a load instruction.

Doshi et al, U.S. Patent No. 6,192,515, has taught a method for software pipelining nested loops. It is further disclosed that after a branch misprediction occurs, which would result in a data hazard, the instructions on the mispredicted path along with their associated results, must be flushed.

Callahan et al., "Software Prefetching," Rice University, 1991, has taught translating a load into a prefetch and a register operation via insertion of a prefetch instruction some time before the register operation.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
July 8, 2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100